LISTING OF THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (currently amended)   A method for developing ~~a~~ an electronic message interface, said method comprising:

using a first software application to develop ~~developing~~ a source document in a first ~~language~~ format, said source document providing instructions for said electronic message interface to enable a second software application operating on a first computer system to communicate ~~directing electronic communication~~ with a third ~~first computer~~ software application operating on a ~~first~~ second computer system, wherein said instructions comply with an application program interface provided by said second software application;

using said first software application to electronically ~~evaluating~~ evaluate said source document to identify keywords, said keywords representing directives ~~in said source document~~ for said electronic message interface ~~communication with said first computer software application~~;

using said first software application to electronically generate ~~generating~~ program source code in a ~~second~~ high level programming language different than said first format as a function of said step of identifying said keywords in said source document, said program source code comprising directives for said electronic message interface ~~communication with said first computer software application~~; and

compiling said program source code into object code, said object code functioning as said electronic message interface comprising said instructions complying with said application program interface ~~to format data messages sent by a second computer system to comply with an application program interface corresponding with said first computer software application~~.

2. (currently amended)   The method of claim 1, wherein ~~further comprising directing~~ said electronic message interface ~~to format~~ formats ~~said~~ a data message ~~messages~~ prior to a middleware application transmitting said data message to said ~~first~~ second computer system.

3. (currently amended)   The method of claim 1, ~~further comprising directing~~ said electronic message interface ~~to format~~ formats ~~said~~ a data message ~~messages~~ after a middleware application transmits said data message to said first computer system.

4. (original)   The method of claim 1, wherein said first computer system is a host system and said second computer system is a client system.

5. (original)   The method of claim 1, wherein said first computer system is a client system and said second computer system is a host system.

6. (original)   The method of claim 1, wherein said step of electronically generating source code comprises evaluating conditions corresponding with said keywords identified in said source document, and generating said program source code as a function of said evaluated conditions.

7. (original)   The method of claim 1, further comprising applying a schema to develop said source document, said schema comprising said keywords.

8. (currently amended)   The method of claim 1, wherein said first software application comprises ~~further comprising operating~~ a front-end software application used to generate said source document.

9. (original)   The method of claim 8, wherein said front-end software application prompts a user of said front-end software application with said schema during development of said source document.

10. (original)   The method of claim 8, wherein said front-end software application notifies said user when said source document contains errors.

11. (currently amended)   A ~~system~~ message interface software application operating on

an information processor for developing [[a]] an electronic message interface, said software application system comprising:

a source document development module, said source document development module enabling a user to develop developed a source document in a first format, language, said source document directing providing instructions for said message interface to enable electronic communication with between a first computer software application operating on a first computer system and a second computer software application operating on a second computer system, wherein said instructions comply with an application program interface provided by said first software application;

a program evaluation module, said program evaluation module electronically evaluating said source document to identify keywords in said source document, said keywords representing directives in said source document for said electronic message interface communication with said first computer software application;

a source code generation module, said source code generation module electronically generating program source code in a second high level programming language different than said first format as a function of identifying said keywords in said source document, said program source code comprising directives for said electronic message interface communication with said first computer software application; and

a program compiler, said program compiler compiling said source code into object code, said object code functioning as said electronic message interface comprising instructions complying with said application program interface to format data messages sent by a second computer system to comply with an application program interface corresponding with said first computer software application.

12. (currently amended) The system message interface software application of claim 11, further comprising a middleware application that transmits transmitting said data messages from said second computer system to said first computer system, said data messages formatted by said electronic message interface prior to said middleware application transmitting said data message to said first computer system.

13. (currently amended) The ~~system~~ message interface software application of claim 11, further comprising a middleware application that transmits ~~transmitting~~ ~~said~~ data messages from said second computer system to said first computer system, said data messages formatted by said electronic message interface after said middleware application transmits said data message to said first computer system.

14. (currently amended) The ~~system~~ message interface software application of claim 11, wherein said first computer system is a host system and said second computer system is a client system.

15. (currently amended) The ~~system~~ message interface software application of claim 11, wherein said first computer system is a client system and said second computer system is a host system.

16. (currently amended) The ~~system~~ message interface software application of claim [[1]] 11, wherein said source code generation module evaluates conditions in statements corresponding with said keywords identified in said source document, and said source code generation module generates ~~generating~~ program source code as a function of said evaluated conditions.

17. (currently amended) The ~~system~~ message interface software application of claim 11, further comprising a schema used to develop said source document, said schema comprising said keywords used in said step of generation of said program source code.

18. (currently amended) The ~~method~~ message interface software application of claim 11, further comprising a front-end software application used to generate said source document.

19. (currently amended) The ~~system~~ message interface software application of claim 18, wherein said front-end software application prompts a user of said front-end software application with said schema during development of said source document.

20. (currently amended) The ~~method~~ <u>message interface software application</u> of claim 18, wherein said front-end software application notifies said user when said source document contains errors.

21. (currently amended) A ~~system~~ <u>software application operable in a memory</u> for developing a message interface, said system comprising:

a program evaluation module, said program evaluation module electronically evaluating a source document, said source document ~~developed~~ <u>provided</u> in a first <u>format</u> ~~language~~ and <u>providing instructions for</u> ~~directing~~ electronic communication ~~with~~ <u>between</u> a first computer software application operating on a first computer system <u>and a second computer software application operating on a second computer system, said instructions directing data messages to be formatted to comply with an application program interface provided by said first computer system</u>;

said program evaluation module identifying keywords in said source document, said keywords representing directives in said source document for said electronic communication with said first computer software application;

a source code generation module, said source code generation module electronically generating program source code in a ~~second~~ <u>high level programming</u> language <u>different than said first format</u> as a function of identifying said keywords in said source document, said source code comprising directives for said electronic communication with said first computer software application; and

a program compiler, said program compiler compiling said source code into object code, said object code functioning as said message interface comprising instructions to format data messages sent by a second computer system<u>, wherein said instructions</u> ~~to~~ comply with an application program interface corresponding with said first computer software application.